Djangomom (Rest, Test, Go)

Djangomom is Backend As Platform, which helps you build, test and deploy Restful web API with out you be writing single line code.

It gives you a user interface to design simple to complex DataBase without writing SQL or python code.
It give you postman testable GET, POST, UPDATE and DELETE API end-points on the go, which returns data in the from of Restful JSON format.
It also give you Django admin panel configured to do Create, Retrieve, Update and Delete (CRUD) operations up on your DataBase.
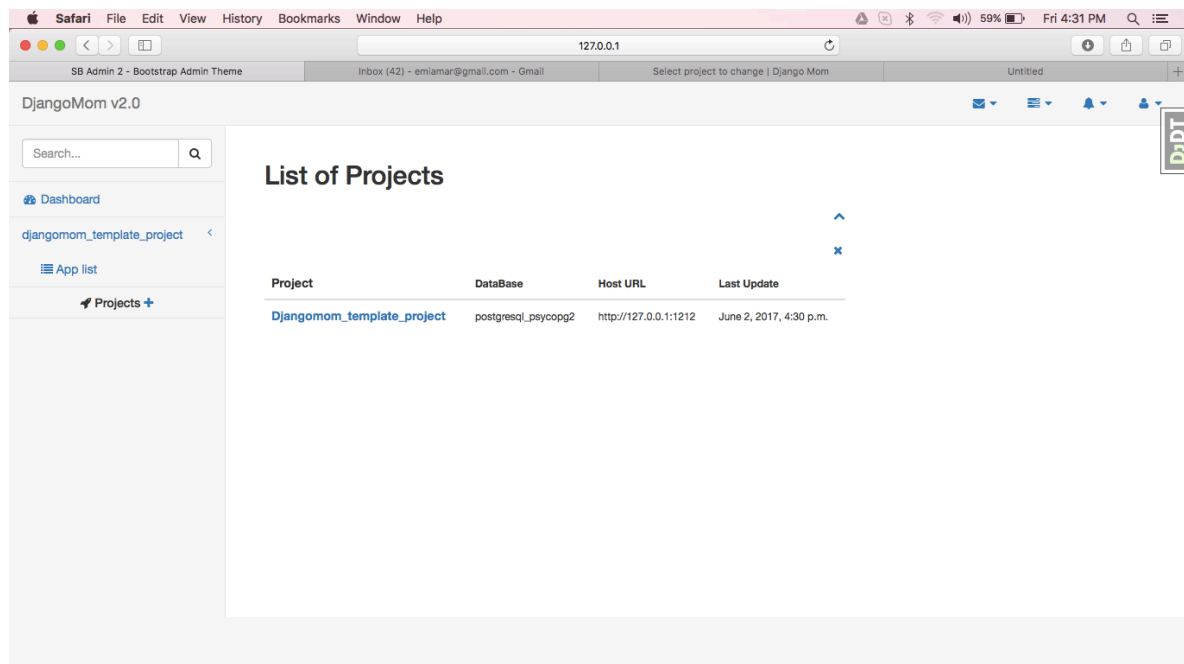You can write business logics, Create custom API endpoints too. Not just limiting user and programmers to UI we allows you to clone the
source code of Django project created in our platform, do changes on you loving editor on you computer and redeploy in lightening speed.
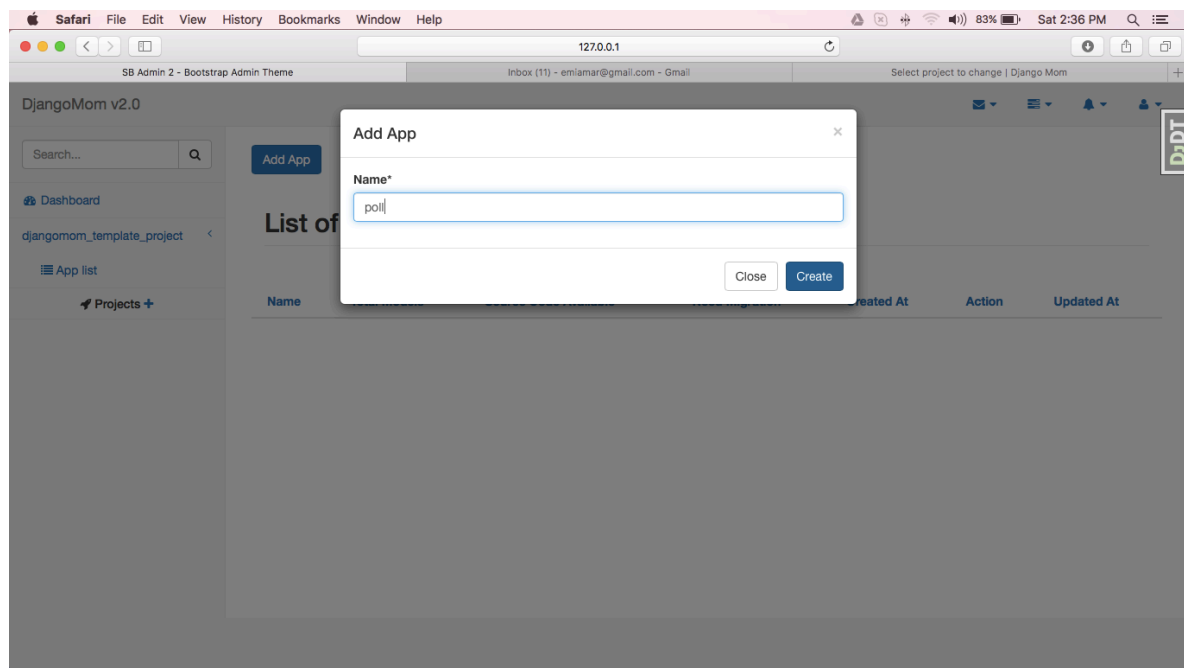
Get Started:

Part 1 :
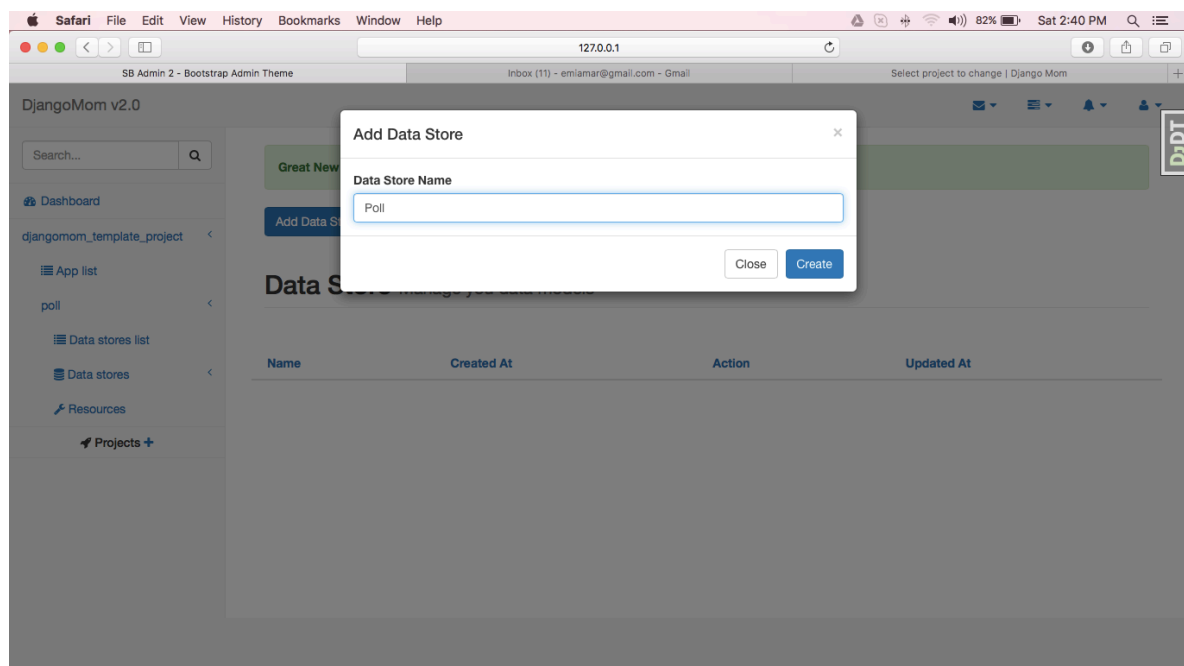Designing Relational Database for django's customary app Polls.

On you /dashboard/ page Create a Project or select existing Project . You will be landed on List of apps page for that particular project.
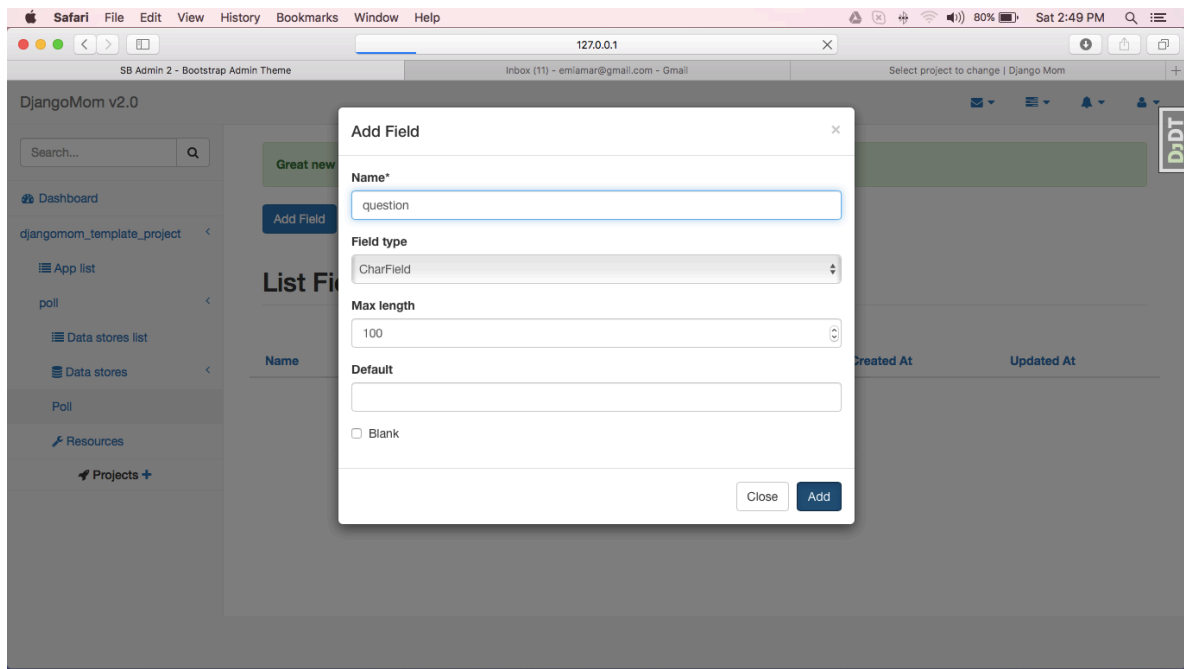

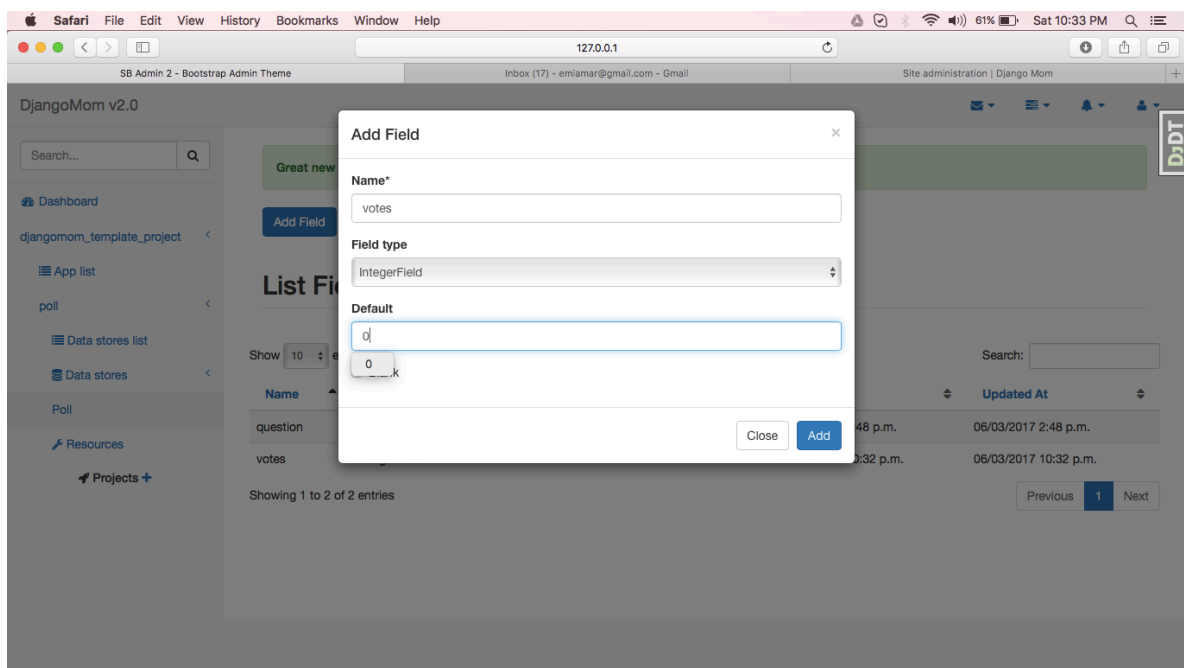Click on Add App, Enter the app name "poll" in the text field.

Great New app "poll" is ready. Lets add new table "Poll" to our store data.
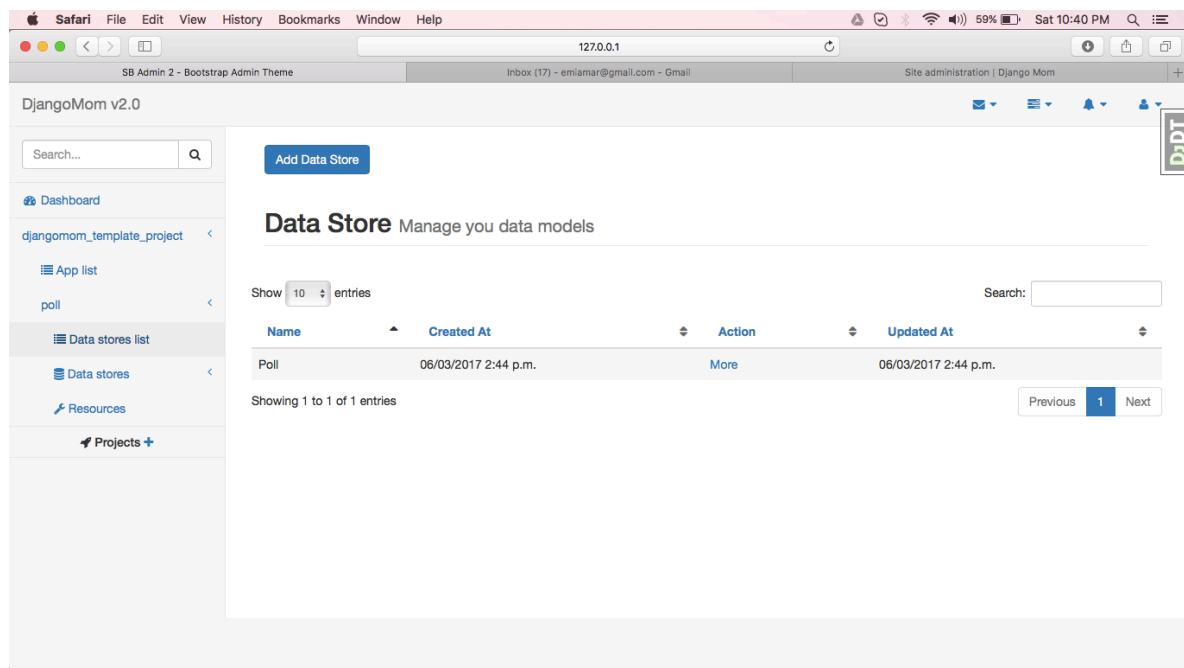

Great now we have created a table called Poll lets start adding field/columns to the database table. Enter name of the field, field type , max length and enter other inputs.
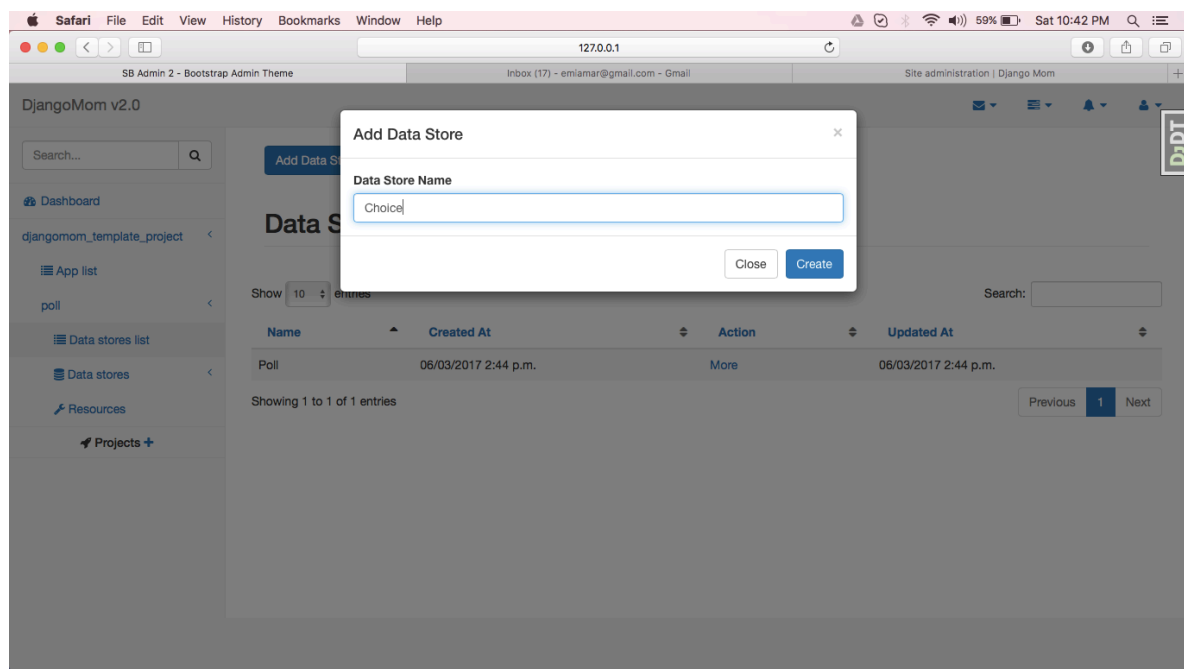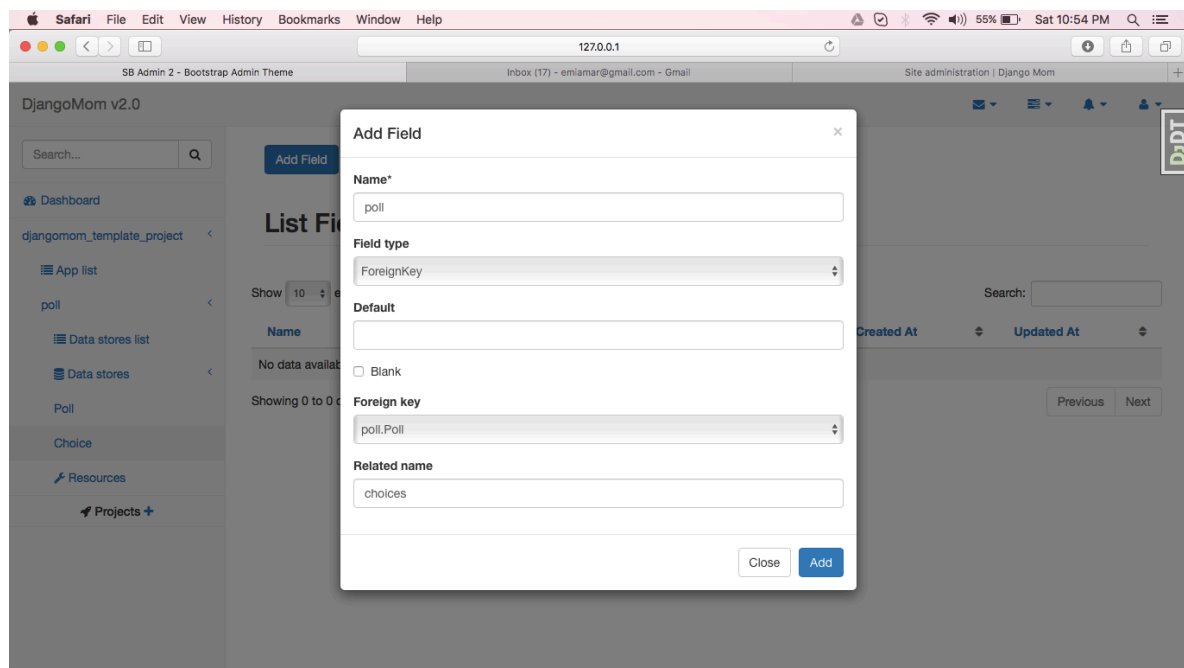
Let's add one more Integer field "votes" to it

Now lets create related table "Choice" to our database Click on Data stores list subtab under poll tab in side menu bar.
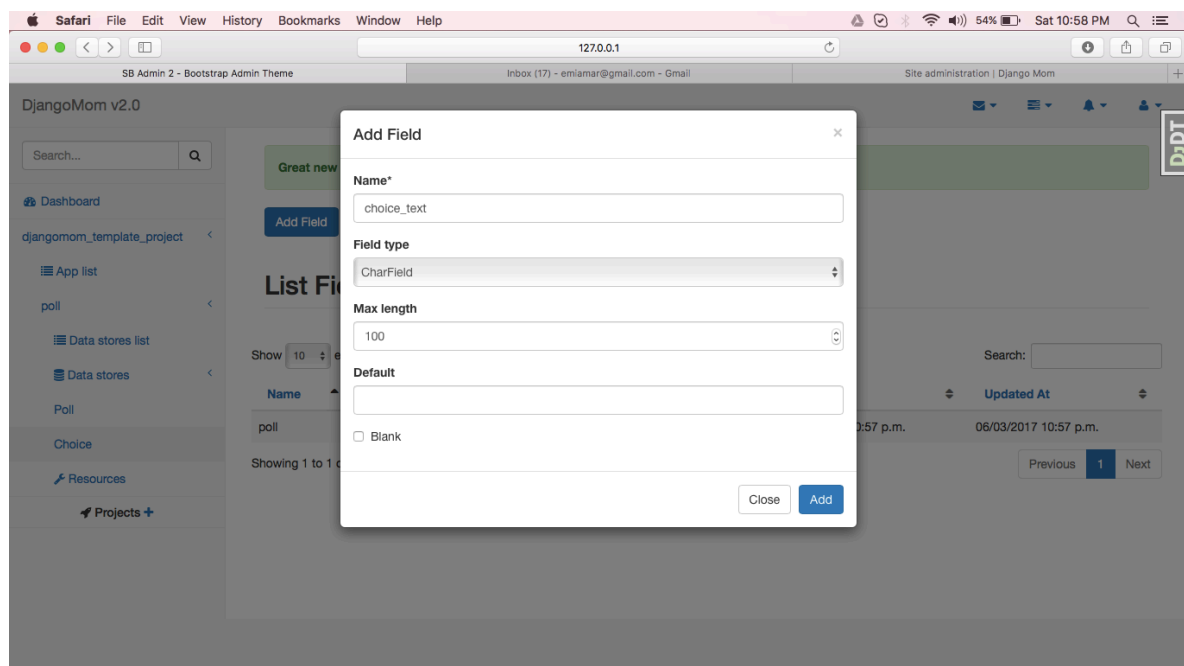
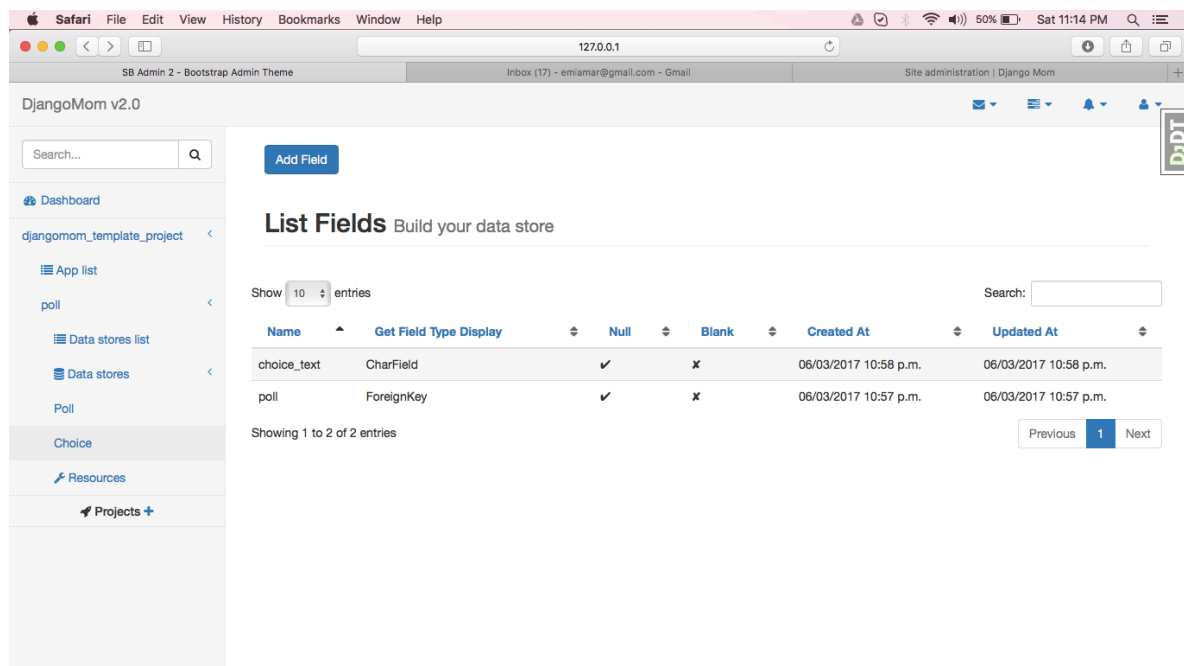Click on Add Data Store tab to add new table "Choice" to your database

Great now we have create new table "Choice" in your database. Let go head and add more fields to it. Since we want relation between "Poll" table and "Choice" table
let create a new field named "poll" which would relates to the "Poll" table. Choose ForeginKey under field type . And select "poll.Poll" under Foreign key dropdown. (You would see drop down items as "appname.TableName"
Example: "poll.Choice" and "poll.Poll") . You can also name the relation which is optional, let call this reaction by name "choices"

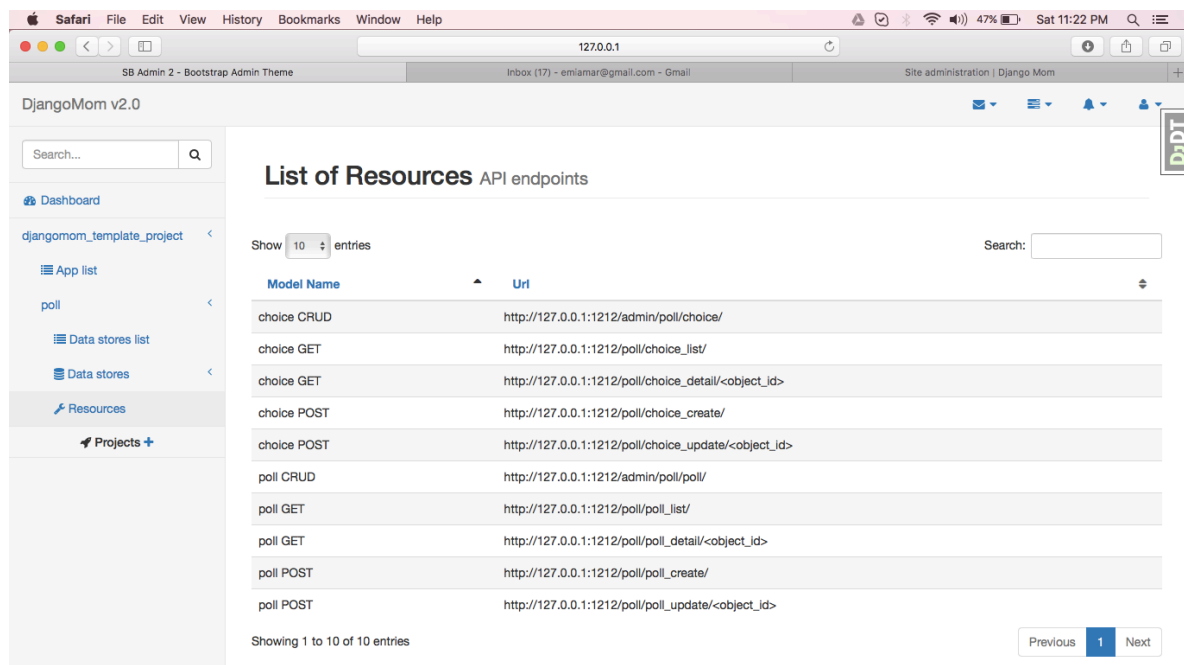Lets add some more fields to store choice text and call it "choice_text" it would be a CharField.

Great now we have an app named "poll" under it we have two data models (tables) named "Poll" and "Choice" with One To Many relationship.
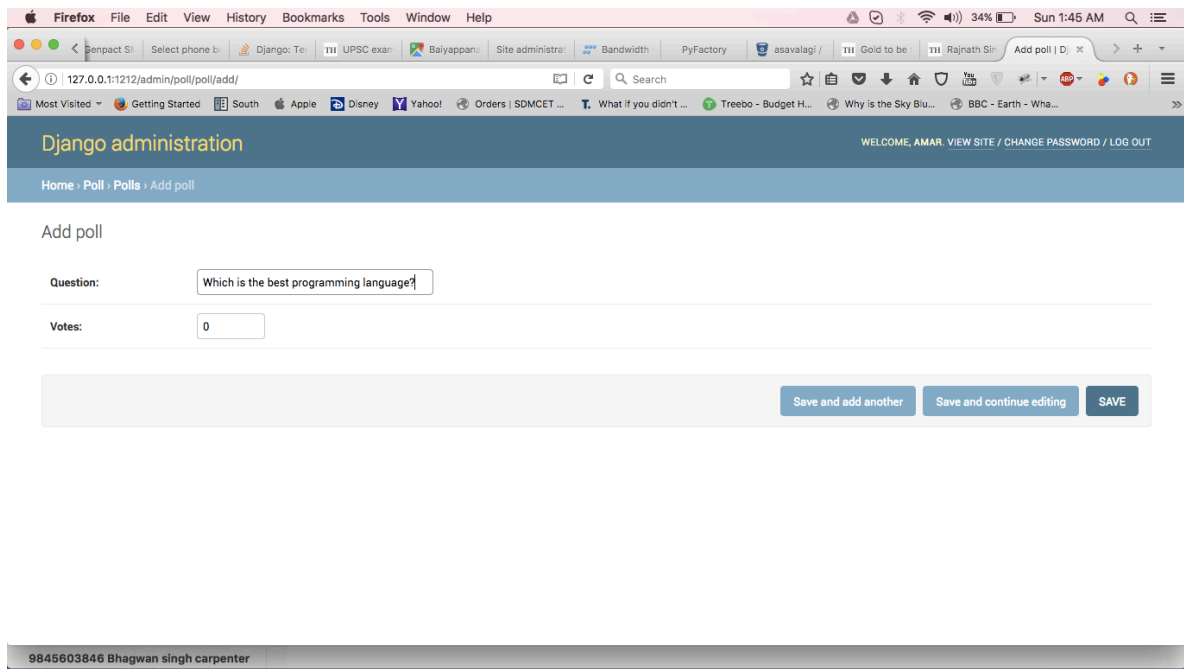
Part 2: Accessing Resources (API endpoints):

Djangomom by default gives you GET, POST, UPDATE and DELETE API endpoint for each of you data models (database tables). Click on "Resources" under poll app. It also gives you django administration panel configured.

Lets click on the 6th url that would take us to "poll" apps CRUD. Add data to "Poll"

Lets